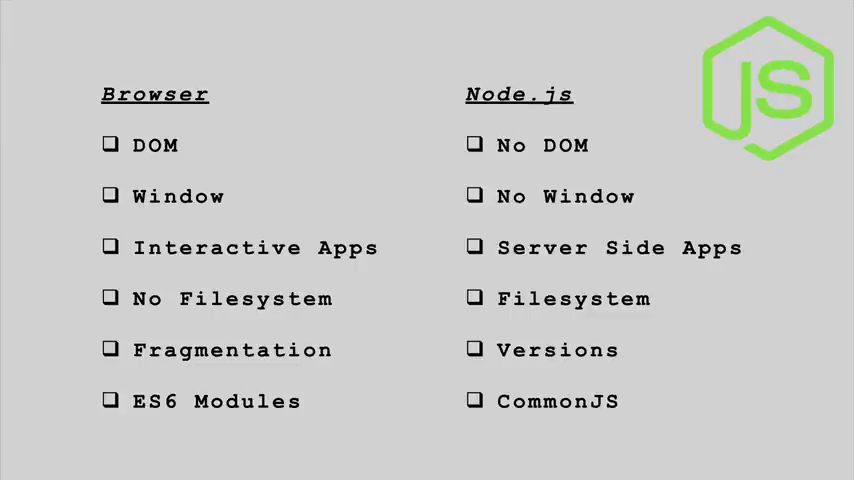
Introduction to Node Js

1. Node Js is an environment to run JS outside the browser. It is built on chrome’s V8 Engine
2. Difference between common and Node JS



1. Globals in Node
   1. Globals in Node are variables that can be accessed anywhere in the entire application
      1. \_\_dirname – path to the current directory
      2. \_\_filename – current filename
      3. Require – function to use modules (just like class)
      4. Process – info about environment where the program is being executed
2. Modules: Just like classes, it is used to group encapsulated code and we share what we want to share from the module. Every file in node is a module.
   1. Module.exports contains list of members of a module that can be shared
   2. To access the members from the above, we use require function

Module.exports = {var1, var2}

e.g const names = require(‘./fileName’)

* 1. The required function is used to create a kind of object of the module
  2. The second example above instantiates the object of ‘fileName’ with a reference variable of names.
  3. Built-in Modules: there are modules that came up with Node JS
     1. OS module provides methods and methods for interacting with the operating system and the server

const os = require(‘os’)

The OS module provides all the information about the current operating system of the device

* + 1. The Path Module deals with the directory of the different files of a particular system folder or files

The path module is useful in building applications in that is to run in different machines and can be very useful.

const filePath = require(‘path’)

* + 1. fileSystem: this provides tools for accessing and manipulating files in the system, the host system. You can read, write and generally manipulate files

The following methods uses synchronous approach – runs code line after the other veto

* + - 1. readFileSync(pathname, methodOfEncode) – this returns the contents of a file in the given pathname, the method of encoding is usually ‘utf8’ for strings
      2. writeFileSync(pathname, file) – this writes the contents of the file to the pathName

Using Asynchronous approach – makes more users access a particular function.

* + - 1. readFile(pathname, methodOfEncode, callBackFunc(error, result)) – the callback function is usually used to get the result and any possible error from the reading process.
      2. writeFile(pathname, methodOfEncode, callBackFunc(error, result))
    1. HTTP module
       1. First we need to import the http module

const http = require(‘http’)

* + - 1. Create a server object – const server = http.createServer((request, response) {})
      2. The createServer method accepts a callback function which provides the request and response objects that is used to communicate to the server from the frontend
      3. The port at which the server will be listening to must be spelt out with the server.listening(portNumber) method
      4. More on http later
    1. npm